

Distributed Estimation of Graph Spectrum*

Mu Yang and Choon Yik Tang

School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK 73019, USA
{muyangwz,cytang}@ou.edu

March 30, 2015

Abstract

In this paper, we develop a two-stage distributed algorithm that enables nodes in a graph to cooperatively estimate the spectrum of a matrix W associated with the graph, which includes the adjacency and Laplacian matrices as special cases. In the first stage, the algorithm uses a discrete-time linear iteration and the Cayley-Hamilton theorem to convert the problem into one of solving a set of linear equations, where each equation is known to a node. In the second stage, if the nodes happen to know that W is cyclic, the algorithm uses a Lyapunov approach to asymptotically solve the equations with an exponential rate of convergence. If they do not know whether W is cyclic, the algorithm uses a random perturbation approach and a structural controllability result to approximately solve the equations with an error that can be made small. Finally, we provide simulation results that illustrate the algorithm.

1 Introduction

The spectrum of a graph, defined as the set of eigenvalues of either its adjacency or Laplacian matrix, provides a useful characterization of the properties of the graph. For instance, the distribution of such eigenvalues offers insights into the shapes and sizes of communities in a complex network [1]. As another example, the largest and smallest of such eigenvalues provides bounds on the maximum, minimum, and average node degrees [2]. The spectrum has also been used, for example, in chemistry, where it is associated with the stability of molecules [2], and in quantum mechanics, where it is related to the energy of Hamiltonian systems [2].

With the continued advances in technology that enable humans to build increasingly complex networks, it is becoming desirable that nodes in a network have the ability to analyze the network themselves, such as decentralizedly computing the spectrum of the network, so that valuable understanding about, say, the network structure may be gained. Motivated by this, a number of

*This work was supported by the National Science Foundation under grant CMMI-0900806.

distributed algorithms have been proposed in the literature, including [3–5] that consider estimation of the entire spectrum of the Laplacian matrix, and [6–8] that focus on estimation of its second smallest eigenvalue (i.e., the algebraic connectivity).

In this paper, we add to the literature by developing a two-stage distributed algorithm, which enables nodes in a graph to cooperatively estimate the spectrum of a matrix W associated with the graph. Unlike in [3–8], the matrix W can be the adjacency or Laplacian matrix of the graph, a weighted version of these matrices, or any other matrix induced by the graph (see Section 2). To construct the algorithm, we first use a discrete-time linear iteration and the Cayley-Hamilton theorem to convert the original problem into an equivalent problem of solving a set of linear equations of the form $Ax = b$, where every row of A and b is known to a particular node (Section 3). We then show that the matrix A can be made almost surely nonsingular if the nodes happen to know that W is cyclic, but not necessarily so if they do not (Section 3). In the case of the former, we use a Lyapunov approach to asymptotically solve the equations with an exponential rate of convergence (Section 4). In the case of the latter, we use a random perturbation approach and a structural controllability result to approximately solve the equations with an error that can be made small (Section 5). Finally, we provide simulation results that illustrate our distributed algorithm (Section 6) and conclude the paper with a word on future research directions (Section 7).

2 Problem Formulation

Consider a network modeled as an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ denotes the set of $N \geq 2$ nodes and $\mathcal{E} \subset \{\{i, j\} : i, j \in \mathcal{V}, i \neq j\}$ denotes the set of edges. Any two nodes $i, j \in \mathcal{V}$ are neighbors and can communicate if and only if $\{i, j\} \in \mathcal{E}$. The set of neighbors of each node $i \in \mathcal{V}$ is denoted as $\mathcal{N}_i = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$, and the communications are assumed to be delay- and error-free, with no quantization.

Suppose associated with the graph \mathcal{G} is a square matrix $W = [w_{ij}] \in \mathbb{R}^{N \times N}$ satisfying the following assumption:

Assumption 1. The matrix W is such that for each $i, j \in \mathcal{V}$ with $i \neq j$, if $\{i, j\} \notin \mathcal{E}$, then $w_{ij} = w_{ji} = 0$.

Note that Assumption 1 allows $w_{ii} \forall i \in \mathcal{V}$ to be arbitrary. It also allows w_{ij} and $w_{ji} \forall \{i, j\} \in \mathcal{E}$ to be arbitrary and different. Thus, W can be the adjacency or Laplacian matrix of graph \mathcal{G} , a weighted version of these matrices, or any other matrix associated with \mathcal{G} as long as Assumption 1 holds.

Suppose each node $i \in \mathcal{V}$ knows only \mathcal{N}_i , w_{ii} , and $w_{ij} \forall j \in \mathcal{N}_i$, which it prefers to not share with any of its neighbors due perhaps to security and privacy reasons. Yet, despite having only such local information about the graph \mathcal{G} and matrix W , suppose every node $i \in \mathcal{V}$ wants to determine

the spectrum of W , i.e., all the N eigenvalues of W , denoted as

$$\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(N)} \in \mathbb{C}, \quad (1)$$

where complex eigenvalues must be in the form of conjugate pairs. Finally, suppose each node $i \in \mathcal{V}$ knows the value of N , which is not an unreasonable assumption since each of them wants to determine the values of N objects.

Given the above, the goal of this paper is to devise a distributed algorithm that enables every node $i \in \mathcal{V}$ to estimate the spectrum (1) of W with a guaranteed accuracy.

3 Forming a Set of Linear Equations

In this section, we show that by having the nodes execute a discrete-time linear iteration N times, the problem of finding the spectrum (1) of W may be converted into one of solving a set of linear equations with appealing properties.

Observe that although none of the nodes has complete information about \mathcal{G} and W , each node $i \in \mathcal{V}$ knows the entire row i of W (since it knows w_{ii} and $w_{ij} \forall j \in \mathcal{N}_i$, and since $w_{ij} = 0 \forall j \notin \{i\} \cup \mathcal{N}_i$ by Assumption 1). This makes the nodes well-suited to carry out the discrete-time linear iteration

$$y_i(t+1) = w_{ii}y_i(t) + \sum_{j \in \mathcal{N}_i} w_{ij}y_j(t), \quad \forall i \in \mathcal{V}, \forall t \in \mathbb{Z}_+, \quad (2)$$

which in matrix form may be written as

$$y(t+1) = Wy(t), \quad \forall t \in \mathbb{Z}_+, \quad (3)$$

where $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$, $y_i(t) \in \mathbb{R}$ is maintained in node i 's local memory, and

$$y(t) = \begin{bmatrix} y_1(t) & y_2(t) & \cdots & y_N(t) \end{bmatrix}^T \in \mathbb{R}^N. \quad (4)$$

Indeed, (2) or (3) can be implemented by having each node $i \in \mathcal{V}$ repeatedly send its $y_i(t)$ to every neighbor $j \in \mathcal{N}_i$.

Since (3) is a discrete-time linear system, we can write

$$y(t) = W^t y(0), \quad \forall t \in \mathbb{Z}_+, \quad (5)$$

so that

$$y(N) = W^N y(0). \quad (6)$$

By the Cayley-Hamilton theorem, W^N in (6) may be expressed as

$$W^N = -x^{(0)}I_N - x^{(1)}W - \cdots - x^{(N-1)}W^{N-1}, \quad (7)$$

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix and the scalars $x^{(0)}, x^{(1)}, \dots, x^{(N-1)} \in \mathbb{R}$ are the N coefficients of the characteristic polynomial of W , i.e.,

$$\begin{aligned} \det(\lambda I_N - W) &= (\lambda - \lambda^{(1)})(\lambda - \lambda^{(2)}) \dots (\lambda - \lambda^{(N)}) \\ &= \lambda^N + x^{(N-1)}\lambda^{N-1} + \dots + x^{(1)}\lambda + x^{(0)}. \end{aligned} \quad (8)$$

Substituting (7) into (6) and using (5), we obtain

$$\begin{aligned} y(N) &= (-x^{(0)}I_N - x^{(1)}W - \dots - x^{(N-1)}W^{N-1})y(0) \\ &= -x^{(0)}y(0) - x^{(1)}y(1) - \dots - x^{(N-1)}y(N-1). \end{aligned} \quad (9)$$

By using (4), we can rewrite (9) as

$$\underbrace{\begin{bmatrix} y_1(0) & y_1(1) & \dots & y_1(N-1) \\ y_2(0) & y_2(1) & \dots & y_2(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ y_N(0) & y_N(1) & \dots & y_N(N-1) \end{bmatrix}}_A \underbrace{\begin{bmatrix} x^{(0)} \\ x^{(1)} \\ \vdots \\ x^{(N-1)} \end{bmatrix}}_{x^*} = \underbrace{\begin{bmatrix} -y_1(N) \\ -y_2(N) \\ \vdots \\ -y_N(N) \end{bmatrix}}_b, \quad (10)$$

where, for later convenience, we denote the matrix on the left-hand side of (10) as $A \in \mathbb{R}^{N \times N}$, the vector of characteristic polynomial coefficients as $x^* \in \mathbb{R}^N$, and the vector on the right-hand side of (10) as $b \in \mathbb{R}^N$.

The matrix equation (10) suggests the following approach for finding the spectrum (1) of W : suppose each node $i \in \mathcal{V}$ selects an initial condition $y_i(0) \in \mathbb{R}$. Upon selecting the $y_i(0)$'s, suppose the nodes execute the discrete-time linear iteration (2) or equivalently (3) N times for $t \in \{0, 1, \dots, N-1\}$. During the execution, suppose each node $i \in \mathcal{V}$ stores the resulting $N+1$ numbers $y_i(0), y_i(1), \dots, y_i(N-1), y_i(N)$ in its local memory. Then, (10) is a set of N linear equations in which each node $i \in \mathcal{V}$ knows the entire row i of A and b , and in which the vector x^* of N characteristic polynomial coefficients $x^{(0)}, x^{(1)}, \dots, x^{(N-1)}$ of W are the N unknowns. It follows that if A is nonsingular, and if the nodes are able to cooperatively solve (10) for the unique x^* , then each of them could determine on its own the N eigenvalues $\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(N)}$ of W using (8) and a polynomial root-finding algorithm.

To realize the above approach, it is necessary that A in (10) is nonsingular. To see whether this can be ensured, observe from (4), (5), and (10) that A may be expressed as

$$A = \begin{bmatrix} y(0) & Wy(0) & \dots & W^{N-1}y(0) \end{bmatrix}. \quad (11)$$

In the form (11), A is, interestingly, the controllability matrix of a fictitious discrete-time single-input linear system

$$z(t+1) = Wz(t) + y(0)u(t), \quad \forall t \in \mathbb{Z}_+, \quad (12)$$

where $z(t) \in \mathbb{R}^N$ is its state, $u(t) \in \mathbb{R}$ is its input, W is its state matrix, and $y(0)$ is its input matrix. Hence:

Proposition 1. *The matrix A in (10) or (11) is nonsingular if and only if the pair $(W, y(0))$ of the system (12) is controllable.*

Since W is given by the problem but $y(0)$ may be freely selected by the nodes, it may be possible to select $y(0)$ so that the pair $(W, y(0))$ is controllable. The following definition and lemmas examine this possibility:

Definition 1 ([9]). A square matrix with real entries is said to be *cyclic* if each of its distinct eigenvalues has a geometric multiplicity of 1.

Lemma 1. *If W is not cyclic, then for every $y(0) \in \mathbb{R}^N$, the pair $(W, y(0))$ is not controllable.*

Proof. Suppose W is not cyclic and let $y(0) \in \mathbb{R}^N$ be given. Then, by Definition 1, W has an eigenvalue $\lambda \in \mathbb{C}$ whose geometric multiplicity exceeds 1, i.e., $\text{rank}(W - \lambda I_N) < N - 1$. Since $y(0)$ is a column vector, $\text{rank}([W - \lambda I_N \mid y(0)]) < N$. Therefore, by statements (i) and (iv) of Theorem 3.1 in [9], the pair $(W, y(0))$ is not controllable. \square

Lemma 2. *If W is cyclic, then for almost every $y(0) \in \mathbb{R}^N$, the pair $(W, y(0))$ is controllable.*

Proof. According to Lemma 3.12 in [9], if $\mathcal{A} \in \mathbb{R}^{n \times n}$ is cyclic and $\mathcal{B} \in \mathbb{R}^{n \times m}$ is such that the pair $(\mathcal{A}, \mathcal{B})$ is controllable, then for almost every $v \in \mathbb{R}^m$, the pair $(\mathcal{A}, \mathcal{B}v)$ is controllable. Applying this lemma with $\mathcal{A} = W$, $\mathcal{B} = I_N$, and $v = y(0)$, and using the fact that the pair (W, I_N) is controllable, we conclude that so is the pair $(W, y(0))$. \square

Proposition 1 and Lemma 1 imply that W being cyclic is necessary for A in (10) or (11) to be nonsingular. Lemma 2, on the other hand, implies that W being cyclic is essentially sufficient because almost every $y(0) \in \mathbb{R}^N$ would work. This latter result is especially useful in a decentralized network because the result allows each node $i \in \mathcal{V}$ to select its $y_i(0) \in \mathbb{R}$ independently from other nodes and randomly from any continuous probability distribution before executing (2) or (3), and be almost sure that the resulting A would be nonsingular.

Motivated by the above analysis, in the rest of this paper we consider separately the following two scenarios:

Scenario 1. The nodes know that W is cyclic.

Scenario 2. The nodes do not know whether W is cyclic, or know that W is not cyclic.

We consider Scenarios 1 and 2 separately because Scenario 1 is easier to deal with (in Section 4) and its treatment helps us deal with Scenario 2 (in Section 5). We note that both of these scenarios arise in applications. For instance, if the graph \mathcal{G} represents a sensor network and the entries w_{ii}

$\forall i \in \mathcal{V}$ and $w_{ij} \forall \{i, j\} \in \mathcal{E}$ of W represent random sensor measurements with continuous probability distributions, then Scenario 1 takes place as the nodes could say with near certainty that W is cyclic because almost every n -by- n matrix has n distinct eigenvalues and, thus, is cyclic. In contrast, if W represents the adjacency or Laplacian matrix of \mathcal{G} , then Scenario 2 takes place as W would be cyclic if \mathcal{G} is, say, a path graph [2] and would not be cyclic if \mathcal{G} is, say, a complete or cycle graph [2], which the nodes could not tell because they only have local information about \mathcal{G} .

To summarize, in this section we have transformed the problem of finding the spectrum (1) of W into one of solving the set of linear equations (10), in which each node $i \in \mathcal{V}$ knows the entire row i of A and b , and in which A can be made almost surely nonsingular in Scenario 1, but not necessarily so in Scenario 2.

4 Solving the Set of Linear Equations in Scenario 1

In this section, we focus on Scenario 1 and develop a continuous-time distributed algorithm that enables the nodes to asymptotically solve the set of linear equations (10) with an exponential rate of convergence.

To facilitate the development, we assume that the nodes have executed (2) or (3) to arrive at (10). Moreover, since A in (10) can be made almost surely nonsingular in this Scenario 1, we assume that it *is* nonsingular throughout the section. With these assumptions, for each $i \in \mathcal{V}$ let $a_i = \begin{bmatrix} y_i(0) & y_i(1) & \cdots & y_i(N-1) \end{bmatrix}^T \in \mathbb{R}^N$ and $b_i = -y_i(N) \in \mathbb{R}$, so that (10) may be stated as

$$\underbrace{\begin{bmatrix} -a_1^T & - \\ -a_2^T & - \\ \vdots & \\ -a_N^T & - \end{bmatrix}}_A \underbrace{\begin{bmatrix} x^{(0)} \\ x^{(1)} \\ \vdots \\ x^{(N-1)} \end{bmatrix}}_{x^*} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}}_b, \quad (13)$$

where a_i and b_i are known to node i because (2) or (3) has been executed. In addition to knowing a_i and b_i , suppose each node $i \in \mathcal{V}$ maintains in its local memory an estimate $x_i(t) = \begin{bmatrix} x_i^{(0)}(t) & x_i^{(1)}(t) & \cdots & x_i^{(N-1)}(t) \end{bmatrix}^T \in \mathbb{R}^N$ of the unknown, unique solution $x^* \in \mathbb{R}^N$, where here $t \in [0, \infty)$ denotes continuous-time (unlike in Section 3 where $t \in \mathbb{Z}_+$ denotes discrete-time). Furthermore, let $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_N(t)) \in \mathbb{R}^{N^2}$ and $\mathbf{x}^* = (x^*, x^*, \dots, x^*) \in \mathbb{R}^{N^2}$ be vectors obtained by stacking the N estimates $x_i(t)$'s and N copies of the solution x^* .

To come up with a distributed algorithm that gradually drives $\mathbf{x}(t)$ to \mathbf{x}^* , consider a quadratic Lyapunov function candidate $V : \mathbb{R}^{N^2} \rightarrow \mathbb{R}$, defined as

$$V(\mathbf{x}) = \sum_{i \in \mathcal{V}} \alpha_i (a_i^T x_i - b_i)^2 + \sum_{\{i,j\} \in \mathcal{E}} \beta_{\{i,j\}} (x_i - x_j)^T (x_i - x_j), \quad (14)$$

where $\alpha_i > 0 \forall i \in \mathcal{V}$ and $\beta_{\{i,j\}} > 0 \forall \{i,j\} \in \mathcal{E}$ are parameters. Notice that each term in the first summation in (14) is a measure of how far away from the hyperplane $\{z \in \mathbb{R}^N : a_i^T z = b_i\}$ the estimate $x_i(t)$ is. Moreover, because A is nonsingular and because of (13), the N hyperplanes $\{z \in \mathbb{R}^N : a_i^T z = b_i\} \forall i \in \mathcal{V}$ have a unique intersection at x^* . Furthermore, the second summation in (14) is a measure of the disagreement among the estimates $x_i(t)$'s. Hence, both the first and second summations in (14) are only positive semidefinite functions of \mathbf{x} . However, as the following proposition shows, adding them up makes V a legitimate Lyapunov function candidate:

Proposition 2. *If A in (13) is nonsingular, then the function V in (14) is positive definite with respect to \mathbf{x}^* .*

Proof. Clearly, V is a positive semidefinite function of \mathbf{x} . To show that it is positive definite with respect to \mathbf{x}^* , we show that $V(\mathbf{x}) = 0$ if and only if $\mathbf{x} = \mathbf{x}^*$. Suppose $\mathbf{x} = \mathbf{x}^*$. Then, $a_i^T x_i - b_i = 0 \forall i \in \mathcal{V}$ according to (13). In addition, the second summation in (14) drops out. Therefore, $V(\mathbf{x}) = 0$. Next, suppose $V(\mathbf{x}) = 0$. Then,

$$a_i^T x_i = b_i, \quad \forall i \in \mathcal{V}, \quad (15)$$

$$x_i = x_j, \quad \forall \{i,j\} \in \mathcal{E}. \quad (16)$$

Since \mathcal{G} is connected, (16) implies that there exists $\tilde{x} \in \mathbb{R}^N$ such that $x_i = \tilde{x} \forall i \in \mathcal{V}$. Substituting this into (15), we get $a_i^T \tilde{x} = b_i \forall i \in \mathcal{V}$ or, equivalently, $A\tilde{x} = b$. Since A is nonsingular, we have $\tilde{x} = x^*$, so that $\mathbf{x} = \mathbf{x}^*$. \square

Remark 1. Notice that V in (14) can also be written as

$$V(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T P (\mathbf{x} - \mathbf{x}^*),$$

where $P = P^T \in \mathbb{R}^{N^2 \times N^2}$ is positive definite and given by

$$P = \begin{bmatrix} \alpha_1 a_1 a_1^T & & & 0 \\ & \alpha_2 a_2 a_2^T & & \\ & & \ddots & \\ 0 & & & \alpha_N a_N a_N^T \end{bmatrix} + L_\beta \otimes I_N,$$

where \otimes denotes the Kronecker product and $L_\beta = [L_{ij}] \in \mathbb{R}^{N \times N}$ is a weighted Laplacian matrix of \mathcal{G} with $L_{ii} = \sum_{j \in \mathcal{N}_i} \beta_{\{i,j\}}$, $L_{ij} = -\beta_{\{i,j\}}$ if $\{i,j\} \in \mathcal{E}$, and $L_{ij} = 0$ if $i \neq j$ and $\{i,j\} \notin \mathcal{E}$. \square

With Proposition 2 in hand, we next take the time derivative of V along the state trajectory $\mathbf{x}(t)$ to obtain

$$\dot{V}(\mathbf{x}(t)) = 2 \sum_{i \in \mathcal{V}} \left[\alpha_i (a_i^T x_i(t) - b_i) a_i + \sum_{j \in \mathcal{N}_i} \beta_{\{i,j\}} (x_i(t) - x_j(t)) \right] \dot{x}_i(t), \quad \forall t \in [0, \infty). \quad (17)$$

Examining (17), we see that $\dot{V}(\mathbf{x}(t))$ can be made negative semidefinite—at the very least—by letting each $\dot{x}_i(t)$ be the negative of the expression within the brackets in (17), i.e.,

$$\dot{x}_i(t) = -\alpha_i(a_i^T x_i(t) - b_i)a_i - \sum_{j \in \mathcal{N}_i} \beta_{\{i,j\}}(x_i(t) - x_j(t)), \quad \forall i \in \mathcal{V}, \forall t \in [0, \infty). \quad (18)$$

The following theorem asserts that the continuous-time system (18) possesses an excellent property:

Theorem 1. *If A in (13) is nonsingular, then the system (18) has a unique equilibrium point at \mathbf{x}^* that is globally exponentially stable, so that $\forall \mathbf{x}(0) \in \mathbb{R}^{N^2}$, $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}^*$, i.e., $\lim_{t \rightarrow \infty} x_i(t) = x^*$ $\forall i \in \mathcal{V}$.*

Proof. For each $i \in \mathcal{V}$, setting $\dot{x}_i(t)$ in (18) to zero yields

$$0 = -\alpha_i(a_i^T x_i - b_i)a_i - \sum_{j \in \mathcal{N}_i} \beta_{\{i,j\}}(x_i - x_j). \quad (19)$$

Summing both sides of (19) over $i \in \mathcal{V}$ gives

$$0 = \sum_{i \in \mathcal{V}} -\alpha_i(a_i^T x_i - b_i)a_i. \quad (20)$$

Due to (13) and to A being nonsingular, the vectors a_1, a_2, \dots, a_N in (20) are linearly independent in \mathbb{R}^N . Thus,

$$0 = -\alpha_i(a_i^T x_i - b_i), \quad \forall i \in \mathcal{V}. \quad (21)$$

Substituting (21) back into (19) results in

$$0 = \sum_{j \in \mathcal{N}_i} \beta_{\{i,j\}}(x_i - x_j), \quad \forall i \in \mathcal{V},$$

which is equivalent to

$$0 = (L_\beta \otimes I_N)\mathbf{x}, \quad (22)$$

where \otimes and L_β have been defined in Remark 1. Since \mathcal{G} is connected, (22) implies that $x_i = \tilde{x}$ $\forall i \in \mathcal{V}$ for some $\tilde{x} \in \mathbb{R}^N$. Plugging this into (21) yields $a_i^T \tilde{x} = b_i$ $\forall i \in \mathcal{V}$. Since A is nonsingular, we have $\tilde{x} = x^*$, i.e., $\mathbf{x} = \mathbf{x}^*$. Hence, the system (18) has a unique equilibrium point at \mathbf{x}^* . Since for each $i \in \mathcal{V}$ the right-hand side of (18) is the negative of the expression within the brackets in (17), $\dot{V}(\mathbf{x}(t))$ is negative definite with respect to \mathbf{x}^* . Therefore, the equilibrium point \mathbf{x}^* is globally exponentially stable. \square

Having established Theorem 1, we now relate it back to the original problem of finding the spectrum (1) of W . To this end, suppose each node $i \in \mathcal{V}$ maintains in its local memory an estimate $\lambda_i^{(\ell)}(t) \in \mathbb{C}$ of the unknown, ℓ th eigenvalue $\lambda^{(\ell)}$ of W for $\ell \in \{1, 2, \dots, N\}$. Also suppose

at each time $t \in [0, \infty)$, node i lets its N estimates $\lambda_i^{(\ell)}(t)$'s be the roots of an N th-order polynomial formed by the estimate $x_i(t) = [x_i^{(0)}(t) \ x_i^{(1)}(t) \ \cdots \ x_i^{(N-1)}(t)]^T$ that is also stored in its local memory, i.e.,

$$\begin{aligned} & (\lambda - \lambda_i^{(1)}(t))(\lambda - \lambda_i^{(2)}(t)) \cdots (\lambda - \lambda_i^{(N)}(t)) \\ &= \lambda^N + x_i^{(N-1)}(t)\lambda^{N-1} + \cdots + x_i^{(1)}(t)\lambda + x_i^{(0)}(t), \quad \forall i \in \mathcal{V}, \forall t \in [0, \infty), \end{aligned} \quad (23)$$

which can be implemented using a polynomial root-finding algorithm embedded in node i . Then, because $(\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(N)})$ in (8) is a continuous function of x^* , and $(\lambda_i^{(1)}(t), \lambda_i^{(2)}(t), \dots, \lambda_i^{(N)}(t))$ in (23) is the *same* continuous function of $x_i(t)$, Theorem 1 implies that

$$\lim_{t \rightarrow \infty} \lambda_i^{(\ell)}(t) = \lambda^{(\ell)}, \quad \forall i \in \mathcal{V}, \forall \ell \in \{1, 2, \dots, N\}. \quad (24)$$

Equation (24), in turn, implies that the system (18) is a continuous-time distributed algorithm that enables the nodes to asymptotically learn the spectrum (1) of W .

Putting together the development in Sections 3 and 4, we obtain the following two-stage distributed algorithm, which is applicable to this Scenario 1:

Algorithm 1 (For Scenario 1).

1. Each node $i \in \mathcal{V}$ selects its $y_i(0) \in \mathbb{R}$ independently from other nodes and randomly from any continuous probability distribution.
2. Upon completion, the nodes execute (2) or (3) N times for $t \in \{0, 1, \dots, N-1\}$, so that each node $i \in \mathcal{V}$ gradually learns the entire row i of A and b in (10).
3. Upon completion, the nodes execute (18) and (23) for $t \in [0, \infty)$, so that each node $i \in \mathcal{V}$ is able to continuously update its $x_i(t)$ and $\lambda_i^{(\ell)}(t)$'s. ■

Remark 2. The current literature offers a few distributed algorithms [10, 11] that may be used to solve linear equations (10). These algorithms are different from (18) in that they force the state of each node to stay in an affine set, whereas (18) allows the state to freely roam the state space. □

5 Solving the Set of Linear Equations in Scenario 2

In this section, we focus on Scenario 2 and provide a slightly different algorithm that enables the nodes to approximately solve (10) with an error that can be made small.

Recall that Scenario 2 represents a situation where the nodes either do not know whether W is cyclic, or somehow know that W is not cyclic. Consequently, they either do not know whether A in (10) is nonsingular, or know that A is singular. Although the nodes could still apply Algorithm 1, there is no guarantee that their estimates $x_i(t)$'s would converge to x^* . One way to address this issue is to have the nodes randomly perturb the matrix W and vector $y(0)$, so that the resulting A in (11) is hopefully nonsingular. Of course, such a random perturbation approach no longer

allows them to asymptotically determine the exact spectrum of W . However, getting an estimate of the spectrum of W may be sufficient in some applications. Thus, we will adopt this random perturbation approach in this Scenario 2.

For notational simplicity, let the matrix associated with the graph \mathcal{G} be denoted as $\overline{W} = [\overline{w}_{ij}] \in \mathbb{R}^{N \times N}$ instead of $W = [w_{ij}]$, and let W instead denote a perturbed version of \overline{W} . In addition, let $\overline{x}^{(\ell)}$'s and $\overline{\lambda}^{(\ell)}$'s denote, respectively, the characteristic polynomial coefficients and eigenvalues of \overline{W} that the nodes wish to determine, and let $x^{(\ell)}$'s and $\lambda^{(\ell)}$'s denote those of W as before. Moreover, let the perturbed matrix W be obtained from \overline{W} in a decentralized manner as follows: prior to executing (2) or (3), each node $i \in \mathcal{V}$ lets

$$w_{ii} = \overline{w}_{ii} + \delta_{ii}, \quad \forall i \in \mathcal{V}, \quad (25)$$

$$w_{ij} = \overline{w}_{ij} + \delta_{ij}, \quad \forall i \in \mathcal{V}, \forall j \in \mathcal{N}_i, \quad (26)$$

where the δ_{ii} 's and δ_{ij} 's are independent, uniformly distributed random variables in the interval $[-a, a]$, so that $a > 0$ represents the perturbation magnitude. Notice that since $\overline{w}_{ij} = 0 \forall i \in \mathcal{V} \forall j \notin \{i\} \cup \mathcal{N}_i$ by Assumption 1,

$$w_{ij} = 0, \quad \forall i \in \mathcal{V}, \forall j \notin \{i\} \cup \mathcal{N}_i \quad (27)$$

as well. Also note that because the nodes are slated to select their $y_i(0)$'s independently and randomly from a continuous probability distribution, there is no need to further randomly perturb these $y_i(0)$'s.

The following lemma uses a structural controllability result to show that the aforementioned approach is effective:

Lemma 3. *If W is as defined in (25)–(27) and $y(0)$ is as defined in Step 1 of Algorithm 1, then A in (11) is almost surely nonsingular.*

Proof. Reconsider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from Section 2. Let $\mathcal{S} = \{(\mathcal{A}, \mathcal{B}) \in \mathbb{R}^{N \times N} \times \mathbb{R}^N : \mathcal{A}_{ij} = 0 \text{ if } i \neq j \text{ and } \{i, j\} \notin \mathcal{E}\}$ and $\mathcal{S}_c = \{(\mathcal{A}, \mathcal{B}) \in \mathcal{S} : (\mathcal{A}, \mathcal{B}) \text{ is controllable}\} \subset \mathcal{S}$. In addition, let $\mathcal{A}^* = \text{diag}(1, 2, \dots, N) \in \mathbb{R}^{N \times N}$ and $\mathcal{B}^* \in \mathbb{R}^N$ be the all-one vector. Then, $(\mathcal{A}^*, \mathcal{B}^*) \in \mathcal{S}$ according to the definition of \mathcal{S} . Moreover, $(\mathcal{A}^*, \mathcal{B}^*) \in \mathcal{S}_c$ because the controllability matrix formed by $(\mathcal{A}^*, \mathcal{B}^*)$ is a Vandermonde matrix that is nonsingular. These two properties of $(\mathcal{A}^*, \mathcal{B}^*)$, along with the definition of structural controllability [12], imply that every $(\mathcal{A}, \mathcal{B}) \in \mathcal{S}$ is structurally controllable. Next, let $(\mathcal{A}, \mathcal{B}) \in \mathcal{S}$ and $\epsilon > 0$ be given. Then, by Proposition 1 of [12], there exists $(\mathcal{A}_c, \mathcal{B}_c) \in \mathcal{S}_c$ such that $\|\mathcal{A} - \mathcal{A}_c\| < \epsilon$ and $\|\mathcal{B} - \mathcal{B}_c\| < \epsilon$. Hence, \mathcal{S}_c is a dense subset of \mathcal{S} . Lastly, note that $(W, y(0)) \in \mathcal{S}$ due to Assumption 1, (25)–(27), and Step 1 of Algorithm 1. Since \mathcal{S}_c is a dense subset of \mathcal{S} , $(W, y(0))$ is almost surely in \mathcal{S}_c . Therefore, by Proposition 1, A in (11) is almost surely nonsingular. \square

As it follows from Lemma 3, by having the nodes perform the extra step described in (25)–(27), the results developed in Sections 3 and 4 become applicable to this Scenario 2. Furthermore, because both the characteristic polynomial coefficients and eigenvalues of a matrix are continuous functions of its entries, by having the nodes decrease the perturbation magnitude a toward zero, the differences between the $x^{(\ell)}$'s and $\lambda^{(\ell)}$'s of W and the $\bar{x}^{(\ell)}$'s and $\bar{\lambda}^{(\ell)}$'s of \bar{W} can be made arbitrarily small, at least in principle. Note, however, that numerical issues may arise when a is too small, or when the resulting A is ill-conditioned. At present, we do not have answers to these numerical issues, and we believe they are important future research directions.

Based on the above, we obtain the following two-stage distributed algorithm for this Scenario 2:

Algorithm 2 (For Scenario 2).

1. Each node $i \in \mathcal{V}$ executes (25)–(27) to obtain a perturbed matrix W .
2. The remaining steps are identical to those of Algorithm 1. ■

6 Simulation Results

In this section, we present two sets of simulation results that demonstrate the effectiveness of Algorithm 1 for Scenario 1 and Algorithm 2 for Scenario 2.

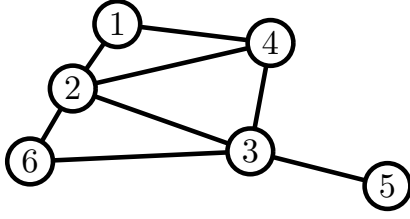
6.1 Simulation of Algorithm 1 for Scenario 1

Consider a sensor network with $N = 6$ nodes, modeled as an undirected, connected graph \mathcal{G} , whose topology is shown in Figure 1(a). Suppose associated with the graph \mathcal{G} is a 6-by-6 matrix W , whose entries satisfy Assumption 1 and represent random sensor measurements given by

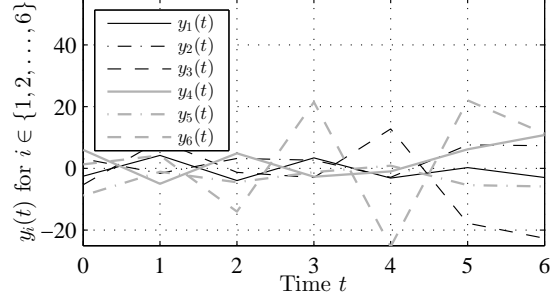
$$W = \begin{bmatrix} -0.10 & -0.24 & 0 & 0.78 & 0 & 0 \\ 0.24 & 0.53 & 0.39 & -0.04 & 0 & -0.19 \\ 0 & 0.34 & 0.21 & 1.15 & -0.13 & 0.71 \\ -0.26 & -0.21 & 0.32 & -0.54 & 0 & 0 \\ 0 & 0 & -0.45 & 0 & 0.39 & 0 \\ 0 & 0.47 & -0.84 & 0 & 0 & -1.35 \end{bmatrix}.$$

Assuming that such measurements are realizations of continuously distributed random variables, the nodes are almost certain that W is cyclic, so that Scenario 1 takes place. Thus, to determine all the eigenvalues $\lambda^{(\ell)}$'s of W , which are given by $-1.02 \pm 0.55i$, $-0.004 \pm 0.46i$, 0.38, and 0.81, the nodes may apply Algorithm 1.

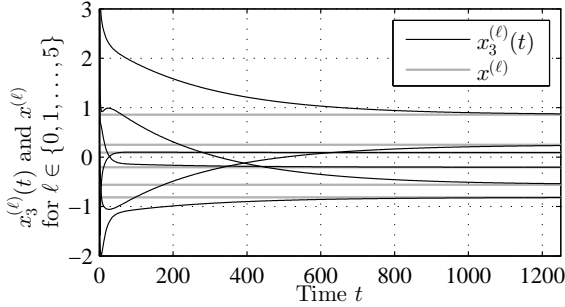
Figures 1(b)–1(d) display the result of simulating Algorithm 1 with $\alpha_i = 10 \forall i \in \mathcal{V}$ and $\beta_{\{i,j\}} = 10 \forall \{i,j\} \in \mathcal{E}$. Specifically, Figure 1(b) shows the data points $y_i(t)$ for $i \in \{1, 2, \dots, 6\}$ and $t \in \{0, 1, \dots, 6\}$ that are used to form the set of linear equations (10). Figure 1(c) shows, as a function of time t , node 3's estimate $x_3^{(\ell)}(t)$ of the ℓ th characteristic polynomial coefficient $x^{(\ell)}$ of



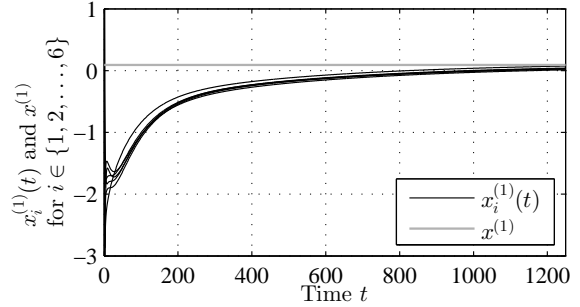
(a) A 6-node graph.



(b) Data points $y_i(t)$ for $i \in \{1, 2, \dots, 6\}$ and $t \in \{0, 1, \dots, 6\}$ that form the set of linear equations (10).



(c) Node 3's estimate $x_3^{(\ell)}(t)$ of the ℓ th characteristic polynomial coefficient $x^{(\ell)}$ for $\ell \in \{0, 1, \dots, 5\}$.



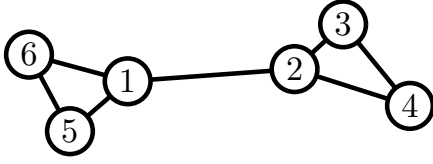
(d) Node i 's estimate $x_i^{(1)}(t)$ of the 1st characteristic polynomial coefficient $x^{(1)}$ for $i \in \{1, 2, \dots, 6\}$.

Figure 1: Performance of Algorithm 1 for Scenario 1.

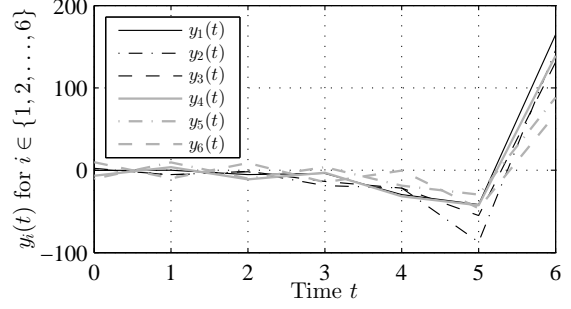
W for $\ell \in \{0, 1, \dots, 5\}$. Likewise, Figure 1(d) shows node i 's estimate $x_i^{(1)}(t)$ of the 1st coefficient $x^{(1)}$ for $i \in \{1, 2, \dots, 6\}$. (Due to space limitation, we are unable to include plots of $x_i^{(\ell)}(t)$ for all $i \in \{1, 2, \dots, 6\}$ and $\ell \in \{0, 1, \dots, 5\}$.) Observe that despite having only local information about \mathcal{G} and W , the nodes are able to utilize Algorithm 1 to asymptotically determine all the characteristic polynomial coefficients $x^{(\ell)}$'s of W and, hence, all its eigenvalues $\lambda^{(\ell)}$'s.

6.2 Simulation of Algorithm 2 for Scenario 2

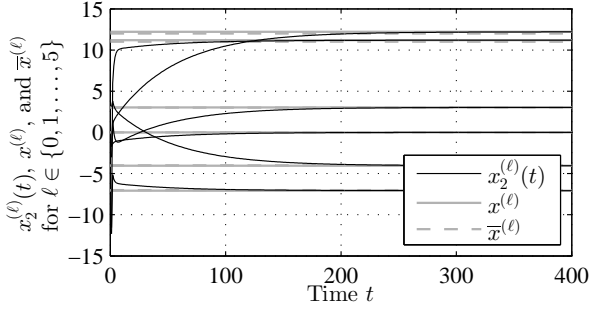
Consider next an undirected, connected graph \mathcal{G} with $N = 6$ nodes, whose topology is shown in Figure 2(a). Let \overline{W} represent the adjacency matrix of \mathcal{G} and suppose the nodes wish to determine all the eigenvalues $\overline{\lambda}^{(\ell)}$'s of \overline{W} , which are given by -1.73 , -1 , -1 , -0.41 , 1.73 , and 2.41 . Because they only have local information about \mathcal{G} , the nodes do not know whether \overline{W} is cyclic, so that Scenario 2 takes place. (In fact, \overline{W} in this particular example is not cyclic because it is symmetric and has repeated eigenvalues, at -1 .) Therefore, the nodes have to apply Algorithm 2. In doing so, they let the perturbation magnitude be $a = 0.2$ and obtain from (25)–(27) a perturbed matrix



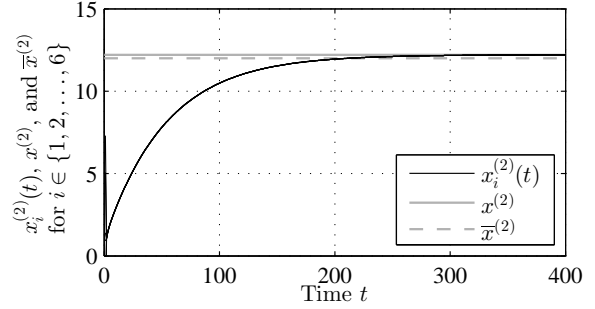
(a) A 6-node graph.



(b) Data points $y_i(t)$ for $i \in \{1, 2, \dots, 6\}$ and $t \in \{0, 1, \dots, 6\}$ that form the set of linear equations (10).



(c) Node 2's estimate $x_2^{(\ell)}(t)$ of the ℓ th perturbed and true characteristic polynomial coefficients $x^{(\ell)}$ and $\bar{x}^{(\ell)}$ for $\ell \in \{0, 1, \dots, 5\}$.



(d) Node i 's estimate $x_i^{(2)}(t)$ of the 2nd perturbed and true characteristic polynomial coefficients $x^{(2)}$ and $\bar{x}^{(2)}$ for $i \in \{1, 2, \dots, 6\}$.

Figure 2: Performance of Algorithm 2 for Scenario 2.

W given by

$$W = \begin{bmatrix} 0 & 1.04 & 0 & 0 & 1.01 & 0.94 \\ 0.98 & 0 & 1.04 & 1.12 & 0 & 0 \\ 0 & 0.98 & 0 & 1.06 & 0 & 0 \\ 0 & 0.95 & 1.01 & 0 & 0 & 0 \\ 0.98 & 0 & 0 & 0 & 0 & 1.01 \\ 0.97 & 0 & 0 & 0 & 0.92 & 0 \end{bmatrix},$$

whose eigenvalues $\lambda^{(\ell)}$'s are -1.74 , -0.97 , -1.03 , -0.40 , 1.73 , and 2.43 , which are all distinct and slightly different from the eigenvalues $\bar{\lambda}^{(\ell)}$'s of \bar{W} .

Figures 2(b)–2(d) display the result of simulating Algorithm 2 with $\alpha_i = 100 \forall i \in \mathcal{V}$ and $\beta_{\{i,j\}} = 10 \forall \{i,j\} \in \mathcal{E}$, using a format similar to that of Figures 1(b)–1(d). The only difference is that Figures 2(c) and 2(d) show not only the characteristic polynomial coefficients $x^{(\ell)}$'s of the “perturbed” W , but also the characteristic polynomial coefficients $\bar{x}^{(\ell)}$'s of the “true” \bar{W} . Observe that with Algorithm 2, the nodes are able to asymptotically determine the $x^{(\ell)}$'s and $\lambda^{(\ell)}$'s. In

other words, they are able to approximately calculate the $\bar{x}^{(\ell)}$'s and $\bar{\lambda}^{(\ell)}$'s with small errors.

7 Conclusion

In this paper, we have designed and analyzed a two-stage distributed algorithm that enables nodes in a graph to cooperatively estimate the graph spectrum. We have shown that asymptotically accurate estimation can be achieved if the nodes know that the associated matrix is cyclic, and estimation with small errors can be achieved if they do not. As for future research, we believe that making the algorithm numerically more robust, so that it can cope with poorly conditioned W and A , is an important next step.

References

- [1] M. E. J. Newman, *Networks: An Introduction*. New York, NY: Oxford University Press, 2010.
- [2] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.
- [3] T. Sahai, A. Speranzon, and A. Banaszuk, “Hearing the clusters of a graph: A distributed algorithm,” *Automatica*, vol. 48, no. 1, pp. 15–24, 2012.
- [4] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, “Decentralized estimation of Laplacian eigenvalues in multi-agent systems,” *Automatica*, vol. 49, no. 4, pp. 1031–1036, 2013.
- [5] T.-M. D. Tran and A. Y. Kibangou, “Distributed estimation of graph Laplacian eigenvalues by the alternating direction of multipliers method,” in *Proc. IFAC World Congress*, Cape Town, South Africa, 2014, pp. 5526–5531.
- [6] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, “Decentralized estimation and control of graph connectivity for mobile sensor networks,” *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.
- [7] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, “Distributed algebraic connectivity estimation for adaptive event-triggered consensus,” in *Proc. American Control Conference*, Montreal, Canada, 2012, pp. 32–37.
- [8] C. Li and Z. Qu, “Distributed estimation of algebraic connectivity of directed networks,” *Systems & Control Letters*, vol. 62, no. 6, pp. 517–524, 2013.
- [9] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall, 1996.

- [10] A. Nedić, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [11] S. Mou and A. S. Morse, “A fixed-neighbor, distributed algorithm for solving a linear algebraic equation,” in *Proc. European Control Conference*, Zurich, Switzerland, 2013, pp. 2269–2273.
- [12] C.-T. Lin, “Structural controllability,” *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.